

# SYLABUS ZAJ /GRUPY ZAJ

## Dane ogólne:

Jednostka organizacyjna:	Katedra Informatyki				
Kierunek studiów:	Informatyka				
Specjalno /Specjalizacja:					
Nazwa zaj / grupy zaj :	Narzędzia i środowiska programistyczne II				
Course / group of courses:	Tools and Programming Environments II				
Forma studiów:	stacjonarne				
Nazwa katalogu:	WP-IN-I-20/21Z				
Nazwa bloku zaj :					
Kod zaj /grupy zaj :	105989	Kod Erasmus:			
Punkty ECTS:	1	Rodzaj zaj :		obowiązkowy	
Rok studiów:	2	Semestr:		4	
Rok	Semestr	Forma zaj	Liczba godzin	Forma zaliczenia	ECTS
2	4	P	15	Zaliczenie z ocen	1
Razem			15		1
Koordynator:	mgr Tomasz Gódek				
Prowadzący zajęcia:					
Język wykładowy:	semestr: 4 - język polski				

## Objaśnienia:

Rodzaj zaj : obowiązkowe, do wyboru.

Forma prowadzenia zaj : W - wykład, - wykład z ćwiczeniami, L - lektorat, S – seminarium/ zajęcia seminaryjne, P - ćwiczenia praktyczne (w tym zajęcia wirtualne), M - ćwiczenia specjalistyczne (medyczne/ kliniczne), LO – ćwiczenia laboratoryjne, LI - laboratorium informatyczne, ZTI - zajęcia z technologii informacyjnych, P – ćwiczenia projektowe, ZT – zajęcia terenowe, T - ćwiczenia terenowe na obozach programowych, SK - samokształcenie (i inne), PR - praktyka zawodowa

## Dane merytoryczne

Wymagania wstępne:			
Znajomość języków i technik programowania, algorytmów i struktur danych przewidzianych programem studiów. Znajomość języka angielskiego w stopniu umożliwiającym studiowanie literatury fachowej.			
Szczegółowe efekty uczenia się			
Lp.	Student, który zaliczył zajęcia, zna i rozumie/potrafi/jest gotowy do:	Kod efektu dla kierunku studiów	Sposób weryfikacji efektu uczenia się
1	Umie projektować i implementować podstawowe funkcje prostej aplikacji z wykorzystaniem różnych technik i języków programowania oraz różnych narzędzi i środowisk wytwarzania oprogramowania.	IN1_U01, IN1_U05	wykonanie zadania
2	Umie posługiwać się wybranymi narzędziami i środowiskami wspierającymi programowanie.	IN1_U05	wykonanie zadania
3	Rozumie trendy rozwoju nowoczesnych technologii, występujących we współczesnej praktyce programowania.	IN1_U11, IN1_U13	wykonanie zadania
4	Umie posługiwać się wybranymi narzędziami wersjonowania oprogramowania oraz narzędziami zarządzania cyklem wytwarzania oprogramowania.	IN1_U13	wykonanie zadania

5	Rozumie trendy rozwoju nowoczesnych technologii, występujących we współczesnej praktyce programowania.	IN1_K01	wykonanie zadania
<b>Stosowane metody osiągnięcia zakładanych efektów uczenia się (metody dydaktyczne)</b>			
metody praktyczne (Podstawowe formy zajęć: ćwiczenia projektowe wymagające samodzielnej analizy zadania projektowego, wykonania projektu, implementacji, testowania i dokumentacji prostej aplikacji lub systemu informatycznego.)			
<b>Kryteria oceny i weryfikacji efektów uczenia się</b>			
<b>umiejętności:</b> ocena wykonania zadania (Projekt)			
<b>kompetencje społeczne:</b> ocena wykonania zadania (Projekt)			
<b>Warunki zaliczenia</b>			
1. Samodzielne przygotowanie projektu oraz implementacji prostej aplikacji lub systemu informatycznego. 2. Przygotowanie i ocena dokumentacji projektowej według podanych założeń.			
<b>Treści programowe (opis skrócony)</b>			
1. Narzędzia i środowiska wytwarzania oprogramowania. 2. Narzędzia wspomagające wytwarzanie aplikacji. 3. Wersjonowanie oprogramowania. 4. Cykle życia oprogramowania.			
<b>Content of the study programme (short version)</b>			
1. Tools and environments for software development. 2. Development tools for applications. 3. Software versioning. 4. Application development life-cycle.			
<b>Treści programowe</b>			
			Liczba godzin
Semestr: 4			
Forma zajęć : <b>ćwiczenia projektowe</b>			
1. Omówienie wybranego zintegrowanego środowiska programistycznego (IDE): Tworzenie projektów, mechanizmy edycyjne, zarządzanie kodem, kompilacja, uruchamianie i debugowanie projektów, refaktoryzacja kodu. 2. Narzędzia wspomagające wytwarzanie aplikacji: Omówienie wybranych platform: Bitbucket, Github lub Gitlab 3. Wersjonowanie oprogramowania: Instalacja i konfiguracja rozproszonego systemu kontroli wersji, podstawowe polecenia, zaawansowane polecenia, integracja systemu kontroli wersji z zintegrowanym środowiskiem wytwarzania oprogramowania IDE. 4. Cykle życia oprogramowania.			15
<b>Literatura</b>			
Podstawowa			
S. Prata, Język C. Szkoła programowania. Wyd. IV, Helion 2016			
W. Gajda, Git. Rozproszony system kontroli wersji, Helion 2013			
Uzupełniająca			

#### Dane jakościowe

<b>Przyporządkowanie zajęć/grup zajęć do dyscypliny naukowej/artystycznej</b>		<b>informatyka techniczna i telekomunikacja</b>
<b>Sposób określenia liczby punktów ECTS</b>		
Forma nakładu pracy studenta (udział w zajęciach, aktywność, przygotowanie sprawozdania, itp.)		Obciążenie studenta [w godz.]
Udział w zajęciach		<b>15</b>
Konsultacje z prowadzącym		<b>0</b>

Udział w egzaminie	0	
Bezpośredni kontakt z nauczycielem - inne	3	
Przygotowanie do laboratorium, wicze , zaj	7	
Przygotowanie do kolokwii i egzaminu	0	
Indywidualna praca własna studenta z literatur , wykładami itp.	0	
Inne	0	
Sumaryczne obciążenie prac studenta	25	
Liczba punktów ECTS		
Liczba punktów ECTS	1	
Zajęcia wymagające bezpośredniego udziału nauczyciela akademickiego	L. godzin	ECTS
	18	0,7
Zajęcia o charakterze praktycznym	L. godzin	ECTS
	25	1,0

1 godz = 45 minut; 1 punkt ECTS = 25-30 godzin

W sekcji 'Liczba punktów ECTS' suma punktów ECTS zajęć wymagających bezporedniego udziału nauczyciela akademickiego i o charakterze praktycznym może się różnić od łącznej liczby punktów ECTS dla zajęć /grup zajęć.